Casting of 1588 Best Master Clock Algorithm into 802.1D RSTP Formalism
Revision 2

Geoffrey M. Garner
Consultant

IEEE 802.1 AVB TG
2008.05.09

gmgarner@comcast.net

Introduction - 1

- Recent discussions in the AVB TG, and 802.1AS D2.0 TG ballot comments, have indicated the need for rapid reconfiguration after a change in topology or grandmaster clock (in an 802.1AS network)
  - The discussions have indicated the desire for this to be scalable, i.e., that there not be any timers whose values depend on network size (or number of hops over which synchronization is transported)
- Related to this, it has been pointed out that the spanning tree chosen by RSTP, for data transport, may not be the optimal spanning tree for transporting synchronization
  - In 802.1AS D2.0, the Best Master Clock Algorithm (BMCA) is invoked at end stations but not at bridges, and a spanning tree must be constructed outside of 802.1AS D2.0
- Reference [1] pointed out that elements of Rapid Spanning Tree Protocol (RSTP) can be used for BMC selection at the nodes, and can construct a spanning tree for synchronization
  - This would be a separate and independent spanning tree from the data spanning tree

Introduction - 2

- The full RSTP of 802.1D is not needed, because 802.1AS is only concerned about transporting synchronization using PTP and 802.11v messages, which are confined to a single hop (single LAN, using 802.1D terminology); there is no need to update a forwarding database, nor ensure that data plane frames do not loop

- In the discussion of [1], it was realized that when RSTP is simplified to do BMC selection, the result is very similar to the IEEE 1588 BMCA
  - Reference [1] describes a framework for mapping the aspects of RSTP that are needed for BMC selection to the IEEE 1588 BMCA
- The present presentation continues the work of [1] and attempts to consider how the detailed RSTP state machines might be simplified and modified to represent a simplification of the IEEE 1588 BMCA

- **This presentation is a work in progress; revision is expected after discussion and review**

Corresponding 1588 and RSTP Concepts - 1

IEEE 1588 Term	RSTP Term
Port State	Port Role (Note 1)
Master port	Designated port
Slave port	Root port
Passive port	Alternate port
(Note 2)	Backup port
Grandmaster	Root
clockIdentity (Note 4)	BridgeId (Note 4)
portNumber (Note 4)	PortId (Note 4)
stepsRemoved (Note 3)	RootPathCost
Announce interval	Hello Time
Announce Receipt Timeout	3*(Hello Time)
clockQuality	(Note 5)
priority1	BridgeID priority bits
priority2	(Note 5)
Announce message	Bridge PDU (BPDU)
Set to 255 in IEEE 1588	maxAge

Note 1: RSTP also defines port state, which is not needed here.

Note 2: Handled in 1588, but no term defined.

Note 3: There has been discussion in the AVB TG of generalizing this to more accurately represent phase error accumulation.

Note 4: See next two slides for more detail

Note 5: No corresponding concept in RSTP

Corresponding 1588 and RSTP Concepts - 2

- clockIdentity and BridgeID – both are 8 bytes
  - The 1588 clockIdentity is either an EUI-64, or an EUI-48 mapped to an EUI-64 using IEEE rules (802.1AS uses the latter)
  - The Bridge ID is a MAC address (EUI-48), with 2 bytes appended
    - These 2 bytes contain 4 bits that represent a settable priority, and 12 bits that represent a locally-assigned system ID extension
  - The settable priority is not needed in 802.1AS because the priority1 and priority2 fields are used for this purpose (described in subsequent slides)
  - The locally-assigned extension likely is not needed because its purpose is mainly to prevent consumption of MAC addresses in a VLAN; however, a time-aware system in 802.1AS will contain a single clock with a single clockIdentity (i.e., will not have separate clocks for different VLANs)
  - In the present presentation, the 1588 clockIdentity is used

Corresponding 1588 and RSTP Concepts - 3

- portNumber and PortID – both are 2 bytes
  - In 1588, the ports are numbered from 1 to N, where N is the number of ports
    - The portNumber is the number, in the range 1, 2, …, N, assigned to the port
  - The PortID in RSTP contains a 12 bit port number
    - A settable 4-bit priority is appended to this that allows the relative priorities of the ports to be managed
  - Use of either scheme in 802.1AS would not impact interoperability of the BMCA with other 1588 (non-802.1AS) networks, because all that is necessary in this algorithm is that the portNumber values for the different ports be unique
    - The only potential interoperability issues would arise with management messages; these would have to allow for the possibility of non-consecutive port numbers
- IEEE 1588 defines a portIdentity, which is the concatenation of clockIdentity and portNumber
  - **Not to be confused with RSTP PortID, which corresponds to 1588 portNumber**

Corresponding 1588 and RSTP Concepts - 4

- clockQuality in IEEE 1588 – consists of:
  - clockClass – characterizes the traceability of a clock, when it is grandmaster
  - clockAccuracy – characterizes the time accuracy of a clock when it is grandmaster
  - OffsetScaledLogVariance – characterizes the frequency stability of a clock, when it is grandmaster

Corresponding 1588 and RSTP Concepts - 5

- stepsRemoved, messageAge, and rootPathCost
  - In 1588, each ordinary and boundary clock maintains the number of hops from the current grandmaster
  - This value is conveyed in the stepsRemoved field of the Announce message
  - stepsRemoved is used for 2 main purposes
    - A simply way of removing best master clock information that is circulating indefinitely (the 'count-to-infinity' problem) (Note: 1588 also has an optional "path-trace" feature for solving this problem, which appends the clocks traversed in a TLV)
    - A cost, used to choose among multiple paths to the same potential grandmaster
- In RSTP, separate variables are used for these two purposes
  - messageAge (a BPDU field) accumulates a time value equal to 1 s for each hope from the potential root
  - rootPathCost accumulates the cost of the path to the root, and is used in the priority vector (see following slides)

Corresponding 1588 and RSTP Concepts - 6

- RSTP distinguishes messageAge and rootPathCost because the cost of the path to the root is not the number of hops to the potential root
  - The cost of each link in RSTP depends on link speed
- There has been discussion that it may be desirable in 802.1AS to use a cost that reflects the phase error accumulation on the path
  - Assuming the phase errors on the successive links are independent of each other, the standard deviation of the accumulated phase error is the square-root of the sum of the squares of the standard deviation of the phase error on each link in the path
  - Assuming the phase errors are unbiased and Gaussian, any desired quantile of phase error is proportional to the standard deviation (with the proportionality constant depending on the specific quantile desired, and properly accounting for both negative and positive values of phase error)
    - Note – this is true for some other distributions, but not for all distributions
  - Therefore, 802.1AS could use a cost that is proportional to the sum of the squares of individual link costs (as pointed out in [1], there is no need to take the square root when comparing costs)
  - If this is done, it will be necessary to choose the link costs

Corresponding 1588 and RSTP Concepts - 7

- For now, this presentation will assume that 802.1AS will use a path cost that is proportional to the sum of the squares of link costs
- It will be assumed this pathCost is accumulated in an Announce message TLV
- It will be possible to configure the individual link costs
- 802.1AS will specify a default link cost, based on
  - Minimum requirement for time stamp accuracy
  - 40 ns phase measurement granularity
  - Minimum requirement for clock frequency stability
- Note: any unknown and fixed component of link asymmetry gives rise to a constant phase offset for a particular path. For a small number of links, it may be desirable to consider the worst-case, which is proportional to the hop count. Whether this is significant depends on the relative magnitude of this error compared to the errors listed in the previous bullet item

Corresponding 1588 and RSTP Concepts - 8

- 802.1AS will accumulate stepsRemoved, in the same manner as 1588, and will discard Announce information after MAX_STEPS_REMOVED hops
  - It must still be decided whether 802.1AS should require a feature similar to the 1588 “path trace” feature, i.e., where the clockIdentities of the successive bridges and end stations traversed are recorded
  - For now, this feature is not described in this presentation

Priority Vector - 1

- In RSTP, the priority vector consists of the following 5 items, with (a) most significant and (e) least significant:
  a) Root BridgeId
  b) RootPathCost
  c) BridgeId of the transmitting bridge
  d) PortId of the port on the transmitting bridge that transmits the message in question
  e) PortId of the port on which the message is received (where relevant)

Priority Vector - 2

- In 802.1AS, the following priority vector is consistent with the 1588 BMCA (with (a) most significant and (i) least significant, subject to discussion on a subsequent slide of the case of comparing the same two clocks)

a) grandmaster priority1

b) grandmaster clockClass

c) grandmaster clockAccuracy

d) grandmaster offsetScaledLogVariance

e) grandmaster priority2

f) grandmasterIdentity (clockIdentity of grandmaster)

g) pathCost, which is a cost proportional to the sum of the squares of link costs (this will be accumulated in a TLV); see earlier discussion of this

h) portIdentity of the transmitting bridge (sourcePortIdentity in Announce message; contains clockIdentity and portNumber)

i) portNumber of the port on which the message is received (where relevant)

- For convenience (and as suggested in [1]), (a) – (f) will be collectively referred to as ‘grandmasterBridgeIdentify’

Comparison of two priority vectors - 1

- When 2 priority vectors are compared in RSTP, they are compared as though they were each single unsigned integers
  - The vector with the smaller numerical value represents the better bridge
- When 2 priority vectors are compared in the IEEE 1588 BMCA, a modification is needed to handle the case where the two vectors represent the same grandmaster clock (i.e., grandmasterIdentity is the same in both vectors)
  - If the two vectors represent different grandmaster clocks, i.e., different values of grandmasterIdentity, they are compared as though they were each single unsigned integers, and the vector with the smaller numerical value represents the better clock (just like RSTP)
  - If the two vectors represent the same grandmaster clock, i.e., grandmasterIdentity is the same in both, then components (a) – (e) are ignored (priority1, clockClass, clockAccuracy, scaledLogVariance, priority2), and the comparison is done on the subvectors consisting of components (g) – (i) (stepsRemoved, portIdentity of transmitting bridge, portNumber of receiving bridge (where relevant))

Comparison of two priority vectors - 2

- The reason the 1588 BMCA ignores priorty1, clockClass, clockAccuracy, scaledLogVariance, and priority2 for the case where the grandmasterIdentities are the same is that this represents a case where a clock is upgraded or downgraded
  - In this case, the values of one or more of these ignored parameters in one of the two vectors are no longer relevant
  - If these parameters were used in the comparison, the algorithm might favor a worse path based on parameters that had changed
    - Note that if the 1588 BMCA did use the RSTP comparison, eventually an Announce message with the updated parameters would be received on the better path, and the spanning tree would be changed to use that path
    - The 1588 approach would avoid having a spanning tree with the worse path in effect for a short time

Priority Vector Calculations - 1

- This material is adapted from subclause 17.6 of 802.1D-2004
- *Port priority vector* (per port)
  - Spanning tree priority vector held for the port when the reception and qualification of any Announce messages, and pending update of information has been completed
    - RSTP uses the term *port priority vector*
  - Port priority vector = {grandmasterBridgeIdentity : pathCost : sourcePortIdentity : portNumber}, where
    - grandmasterBridgeIdentity is comprised of items (a) – (f) on slide 13 for the current grandmaster
    - pathCost is the accumulated path cost (sum of squares of link costs) to the current grandmaster
    - sourcePortIdentity is the portIdentity of the master port for the communication path (LAN) that this port is attached to
    - portNumber is the portNumber of this port

Priority Vector Calculations - 2

- *Message priority vector (*per Announce message, per port)
  - Spanning tree priority vector conveyed in a received Announce message
    - RSTP uses the term *message priority vector*
  - For a bridge B receiving an Announce message on a port P_B from a master port P_M on bridge M claiming a grandmasterBridgeIdentity R_M and a path cost PC_M:
    - Port priority vector = {R_M : PC_M : P_M : PN_B}, where
      - R_M: grandmasterBridgeIdentity, comprised of items (a) – (f) on slide 13, for the grandmaster indicated in the received Announce message
      - PC_M: pathCost is the accumulated path cost (sum of squares of link costs) carried in the Announce message TLV
      - P_M: sourcePortIdentity field of the Announce message (it is comprised of clockIdentity and portNumber of the port that sent the Announce message)
      - PN_B: portNumber of the port of this bridge that received the Announce message

Priority Vector Calculations - 3

❑ *GM path priority vector* (per port)

- The GM path priority vector can be calculated from the port priority vector of the receiving port by adding the receiving port's path cost PPC_{PB} to the pathCost component
  - GM path priority vector = $\{R_M : (PC_M)^2 + (PPC_{PB})^2 : P_M : PN_B\}$, where
    - R_M: grandmasterBridgeIdentity, comprised of items (a) – (f) on slide 13, for the receiving port
    - PC_M: pathCost is the accumulated path cost (sum of squares of link costs) to the GM, for the receiving port, excluding the immediate upstream link
    - PPC_{PB}: cost of the immediate upstream link
    - P_M: portIdentity of the master port for the communication path (LAN)
    - PN_B: portNumber of this port
  - RSTP uses the term *root path priority vector*

Priority Vector Calculations - 4

- *Bridge priority vector* (per bridge)
  - The bridge priority vector for a bridge B is the priority vector that would, with the portNumber of the sourcePort Identity set equal to the portNumber of the transmitting master port, be used as the message priority vector in Announce messages transmitted on B's master ports if B were selected as the grandmaster
    - GM path priority vector = {R_B : 0: {C_B : 0 } :0}, where
      - R_B: bridgeIdentity of this bridge
      - C_B: clockIdentity of this bridge
      - {C_B : 0} is a portIdentity whose clockIdentiy is C_B and portNumber is 0
    - RSTP uses the term *bridge priority vector*

Priority Vector Calculations - 5

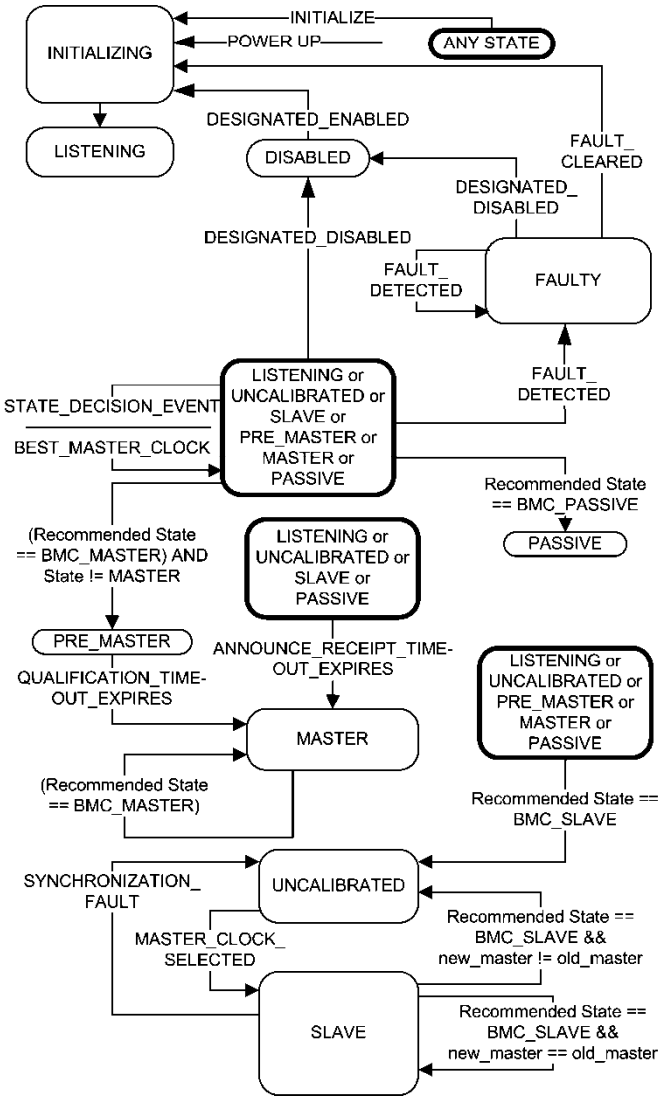
❑ *GM priority vector* (per bridge)

- The GM priority vector for a bridge B is the best of the set comprising the bridge priority vector plus all GM path priority vectors, for whom the clockIdentity of the master portIdentity is not the clockIdentity of B. If the bridge priority vector is best, B has been selected as the grandmaster. For the case where the best GM path priority vector is that of port PN_B in the above GM path priority vector example, then:
  - GM priority vector = $\{R_B : 0: \{C_B : 0\} :0\}$
    - if R_B (bridgeIdentity of the bridge priority vector) is better than R_M (bridge identity of the GM path priority vector)
  - GM priority vector = $\{R_M : (PC_M)^2 + (PPC_{PB})^2: P_M : PN_B\}$
    - if R_B is worse than R_M
  - RSTP uses the term *root priority vector*

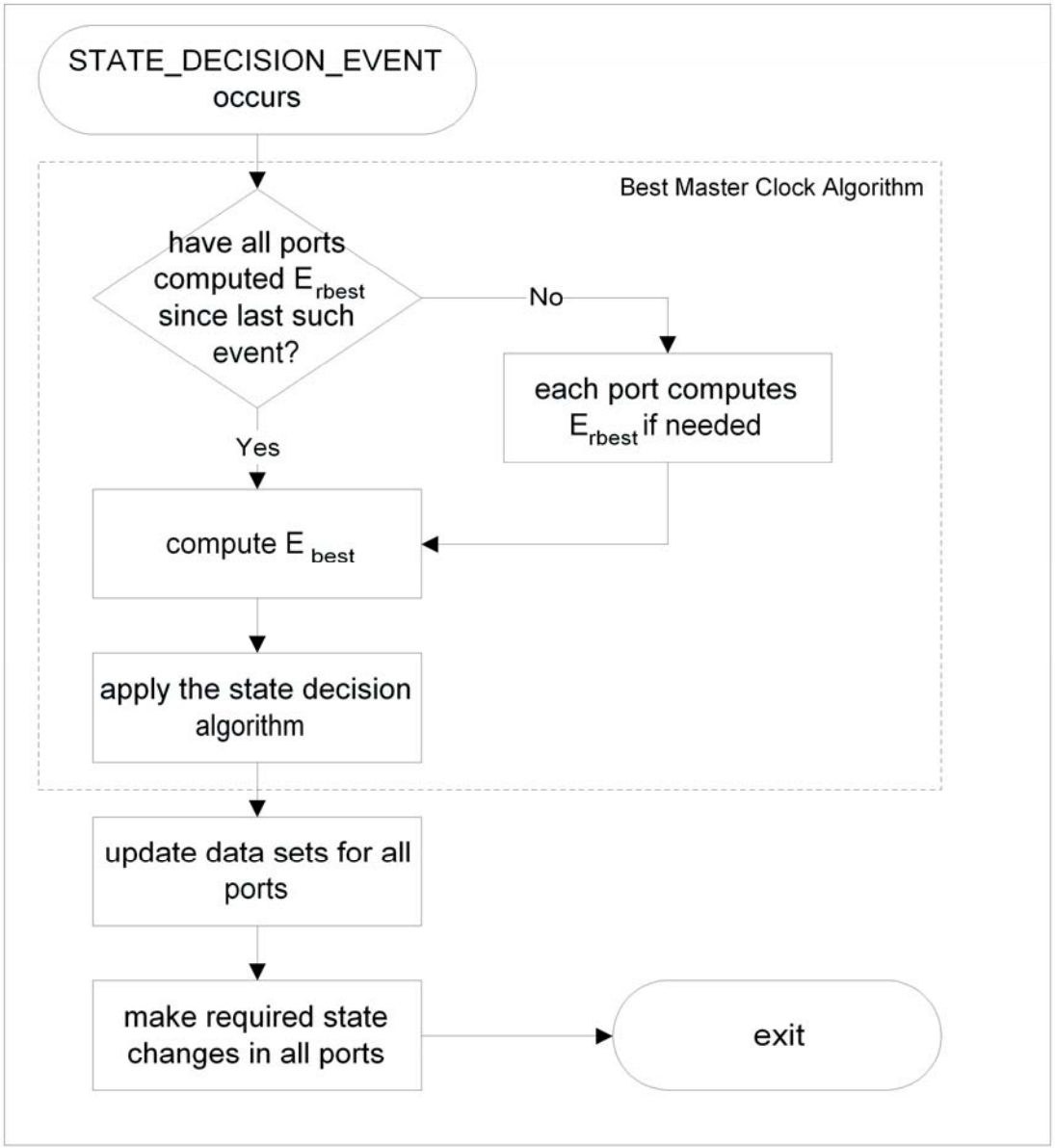
Priority Vector Calculations - 6

- *Master priority vector* (per port)
  - The master priority vector for a port Q on bridge B is the root priority vector with B's clockIdentity C_B substituted for the clockIdentity of the master port portIdentity, and Q's portNumber QB substituted for the master port portNumber and the portNumber of the sending port
    - Master priority vector = $\{R_B : 0: \{C_B : Q_B \} : Q_B\}$
      - –if R_B (bridgeIdentity of the bridge priority vector) is better than R_M (bridge identity of the GM path priority vector)
    - Master priority vector = $\{R_M : (PC_M)^2 + (PPC_{PB})^2: \{C_B : Q_B \} : Q_B\}$
      - –if R_B is worse than R_M
    - RSTP uses the term *designated priority vector*

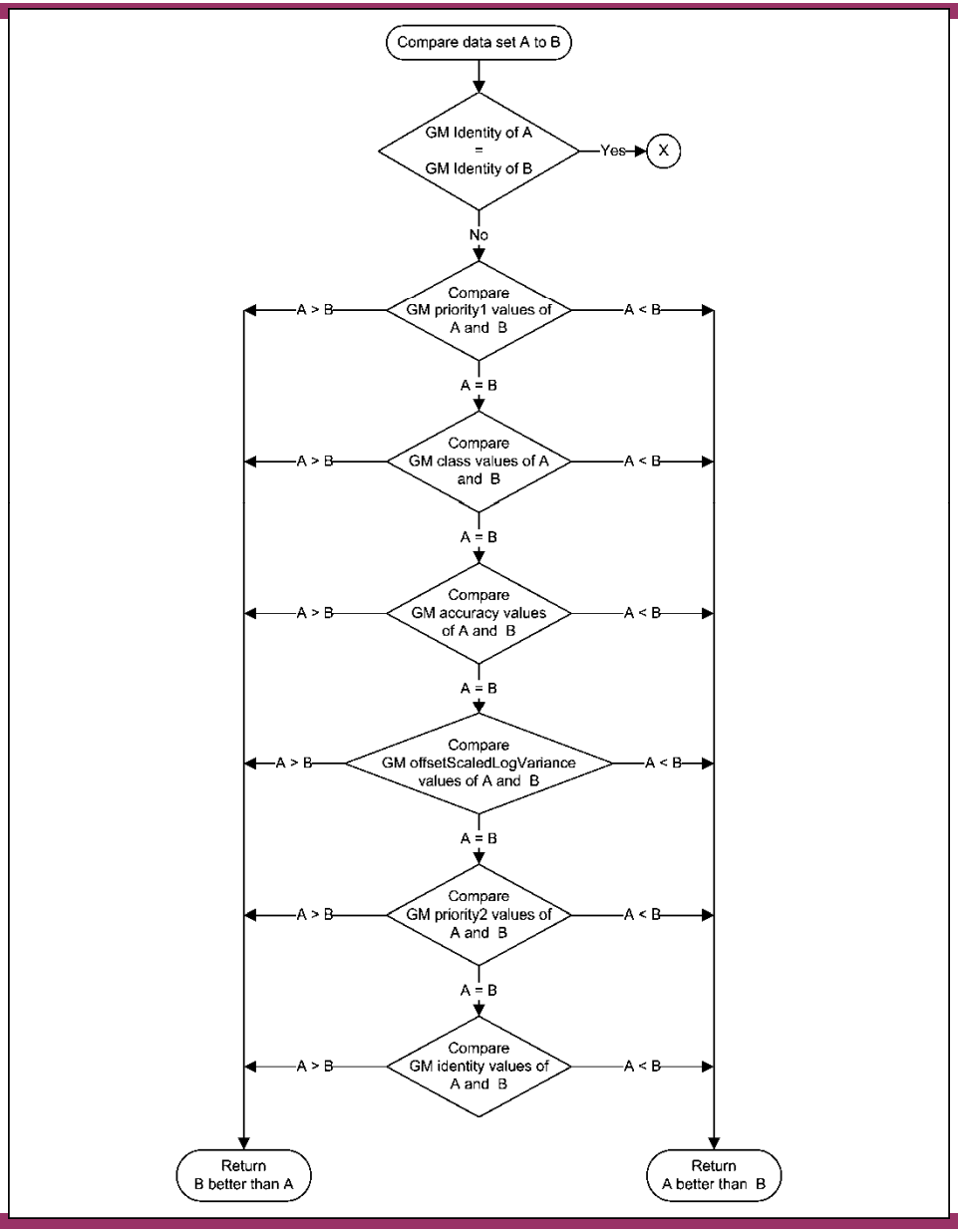
1588 Port State Machine



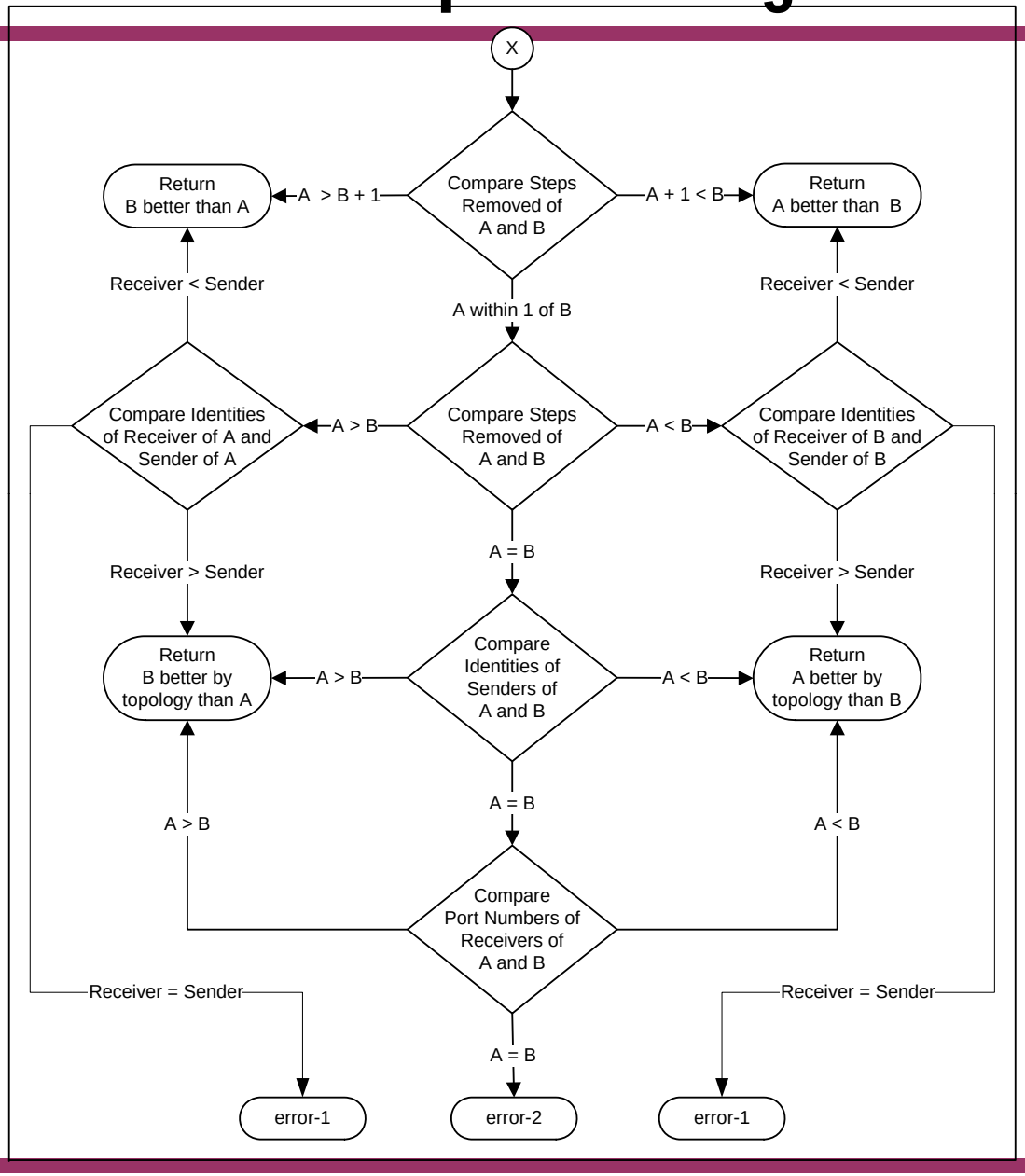
1588 State Decision Event Logic



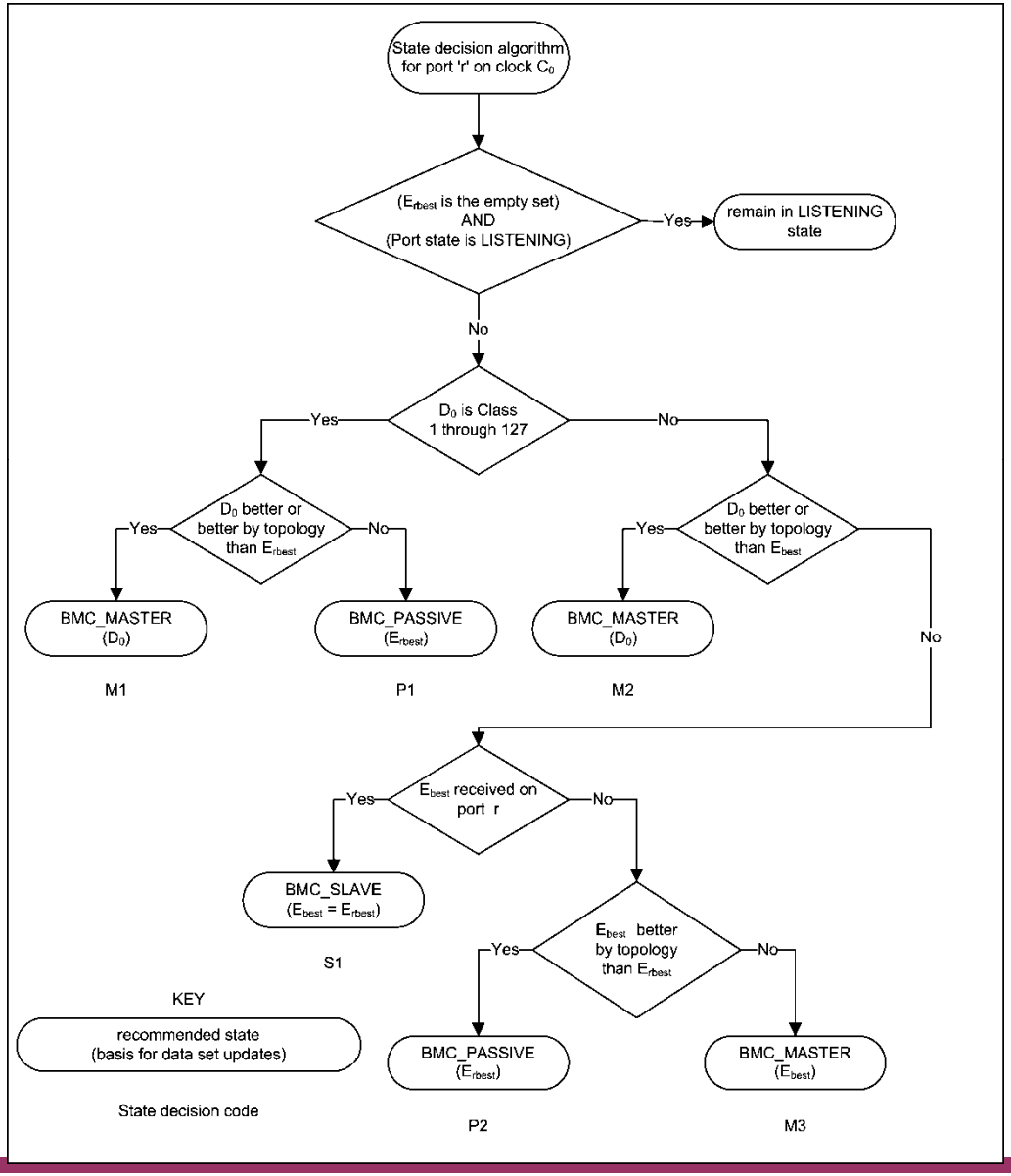
1588 Data Set Comparison Algorithm – 1



1588 Data Set Comparison Algorithm – 2



1588 State Decision Algorithm



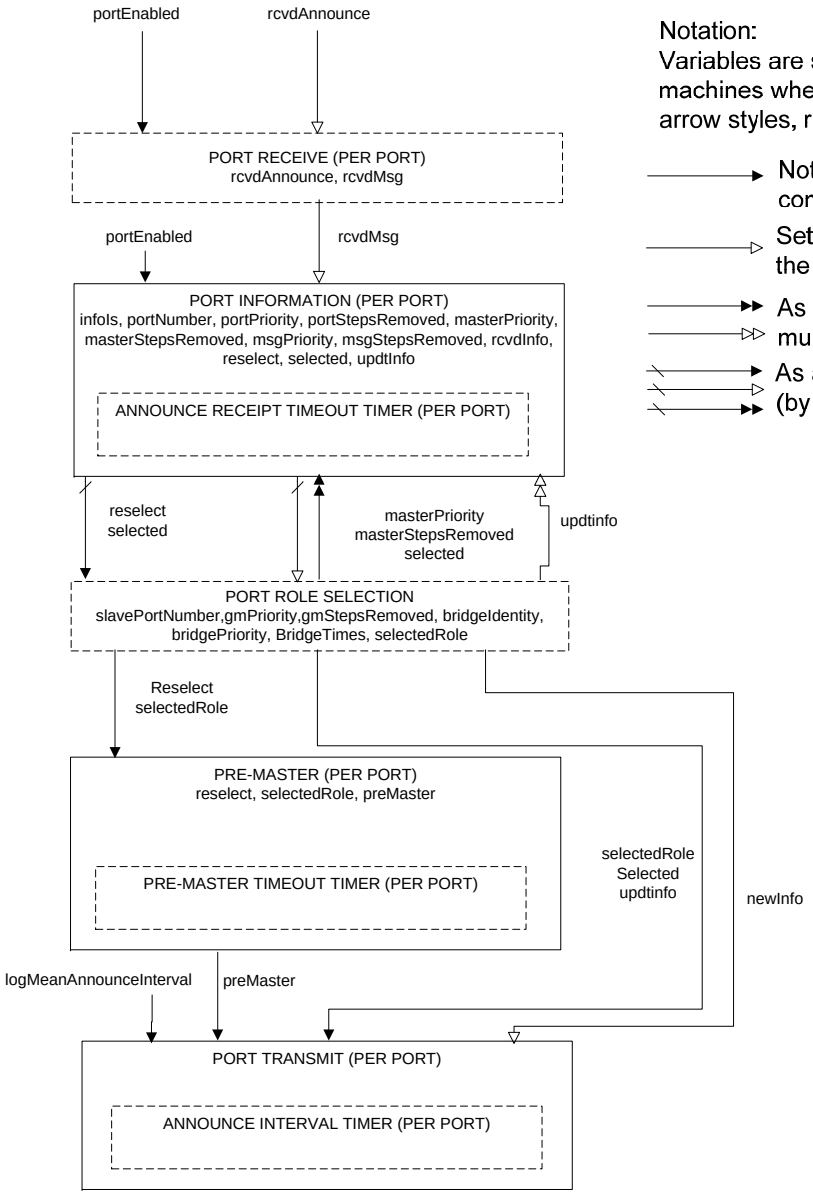
RSTP State Machines Needed for 1588 BMCA - 1

- In the following slides, the RSTP state machines needed to represent the 1588 BMCA are described
  - The state machines are simplified where appropriate
  - RSTP nomenclature is converted to 1588/802.1AS nomenclature (e.g., BPDU becomes Announce message)
  - One new state machine is needed
- The state machines needed include
  - Overview and interrelationships (802.1D/Figure 17-12)
  - Port Receive state machine (802.1D/Figure 17-14)
  - Port Transmit state machine (802.1D/Figure 17-17)
  - Port Information state machine (802.1D/Figure 17-18)
  - Port Role Selection state machine (802.1D/Figure 17-19)
  - Pre-master state machine (new)

RSTP State Machines Needed for 1588 BMCA - 2

- Other RSTP state machines, having to do with detecting the edge of the network (this will be done using the Pdelay mechanism in 802.1AS), updating the forwarding data base in a way that does not cause loops, notifying of topology changes, and compatibility between RSTP and STP, are not needed for the BMCA
- The port timers state machine is eliminated, because 802.1AS uses an equivalent means of modeling a timer
  - 802.1AS assumes that each node has a free-running timer that constantly indicates time in the variable currentTime
  - If it is desired to set a timer, the time at which that timer times out (timeoutTime) is set equal to currentTime plus the timeout value
  - A true value for the condition currentTime >= timeoutTime indicates that the timer has timed out
- For now, the capitalization of the first letter of various variables (or lack thereof) is ignored; this notation will be made consistent in a later version of this presentation and in the next 802.1AS draft

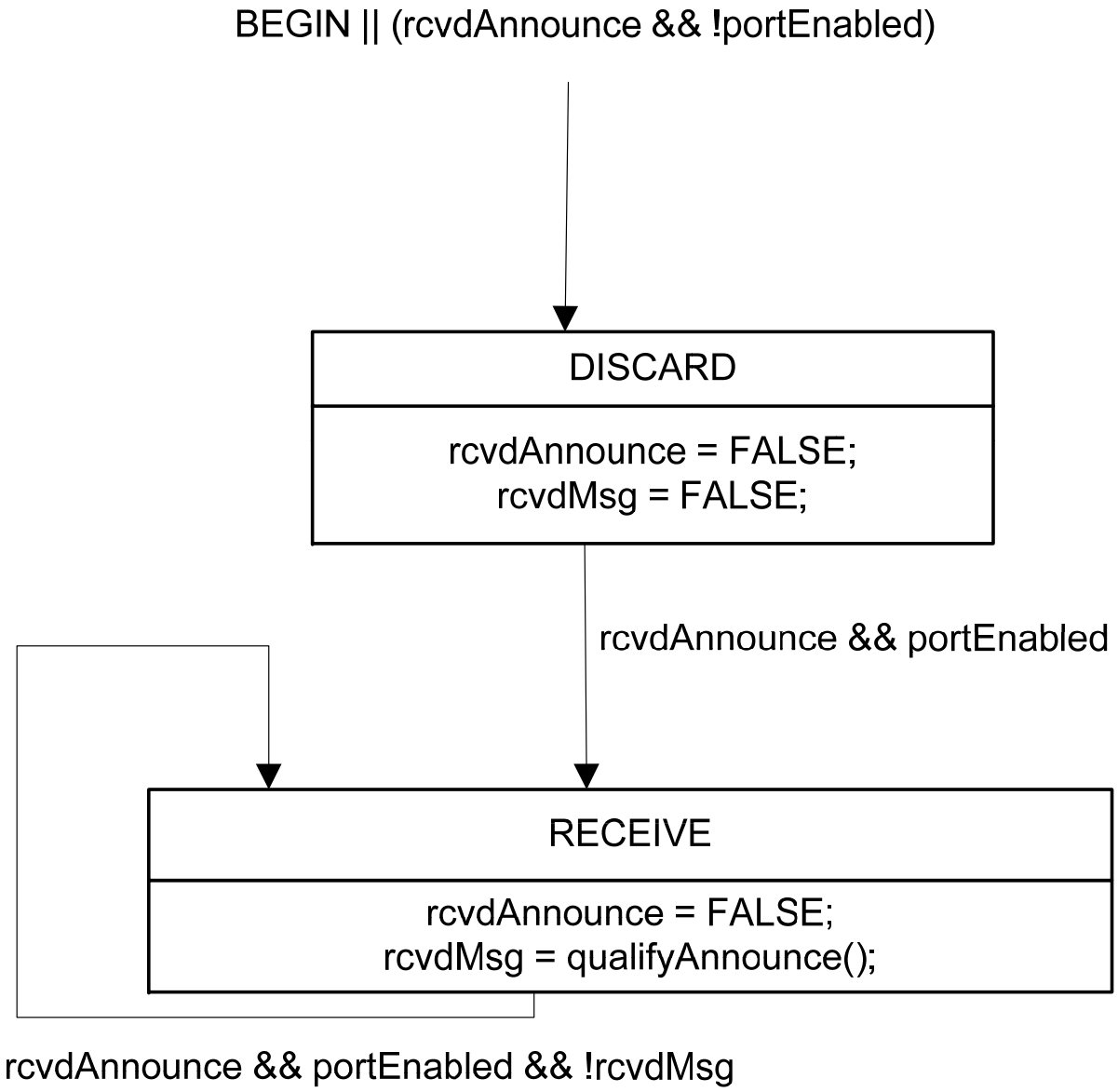
Overview and Interrelationships of state machines



Notation:
Variables are shown both within the machine where they are principally used and between machines where they are used to communicate information. In the latter case a variety of arrow styles, running from one machine to another, show how each is typically used:

- Not changed by the target machine. Where the machines are both per Port, this variable communicates between instances for the same port
- Set (or cleared) by the originating machine, cleared (or set) by the target machine. Where the machines are both per Port, this communicates between instances for the same port.
- As above, except that the originating per port machine instance communicates with multiple port machine instances (by setting or clearing variables owned by those ports).
- As above, except that multiple per Port instances communicate with (an)other instance(s) (by setting or clearing variables owned by the originating ports).

Port Receive State Machine



qualifyAnnounce() function - 1

- Qualifies a received Announce message in accordance with IEEE 1588 requirements
- Each port of an IEEE 1588 OC or BC maintains a "foreign master data set", which contains one entry for each other OC or BC that it has received Announce messages from. Each entry contains
  - The portIdentity (clockIdentity, portNumber) of the port that sent the most recent Announce message from that OC or BC
  - The number of Announce messages received in a time window of duration FOREIGN_MASTER_TIME_WINDOW
- A received Announce message will be qualified *only if* (this is one condition; others will be indicated shortly) FOREIGN_MASTER_THRESHOLD Announce messages have been received from that clock within FOREIGN_MASTER_TIME_WINDOW
- In IEEE 1588, these values are set to:
  - FOREIGN_MASTER_TIME_WINDOW = 4 announce intervals
  - FOREIGN_MASTER_THRESHOLD = 2

qualifyAnnounce() function - 2

- Each port of an 802.1AS bridge or end station will maintain this data set
  - For now, it is also referred to as a “foreign master data set”, though a different name can be chosen if desired
  - This data set is simply a list of portIdentities for distinct bridges or end stations, other than this bridge or end station, that this port has heard from
    - **Note that these are potential masters that the bridge or end station has heard from, not potential grandmasters**
    - For a point-to-point link, there is only one potential master that the port can hear from, because there is only one other endpoint of the link
    - Note that this check serves a different purpose from announce receipt timeout
      - Announce receipt timeout pertains to a bridge or end station that already is master and has gone away
      - The foreign master qualification pertains to a bridge or end station newly heard from
  - Since FOREIGN_MASTER_THRESHOLD = 2, it is not necessary to maintain an explicit field with the number of Announce messages received
    - Absence of an entry indicates 0 Announce messages have been received from this foreign master, and the current Announce message is the first
    - Presence of an entry indicates 1 Announce message has been received from this foreign master, and the current Announce message is the second (and it need only be checked if it is received within the time window)

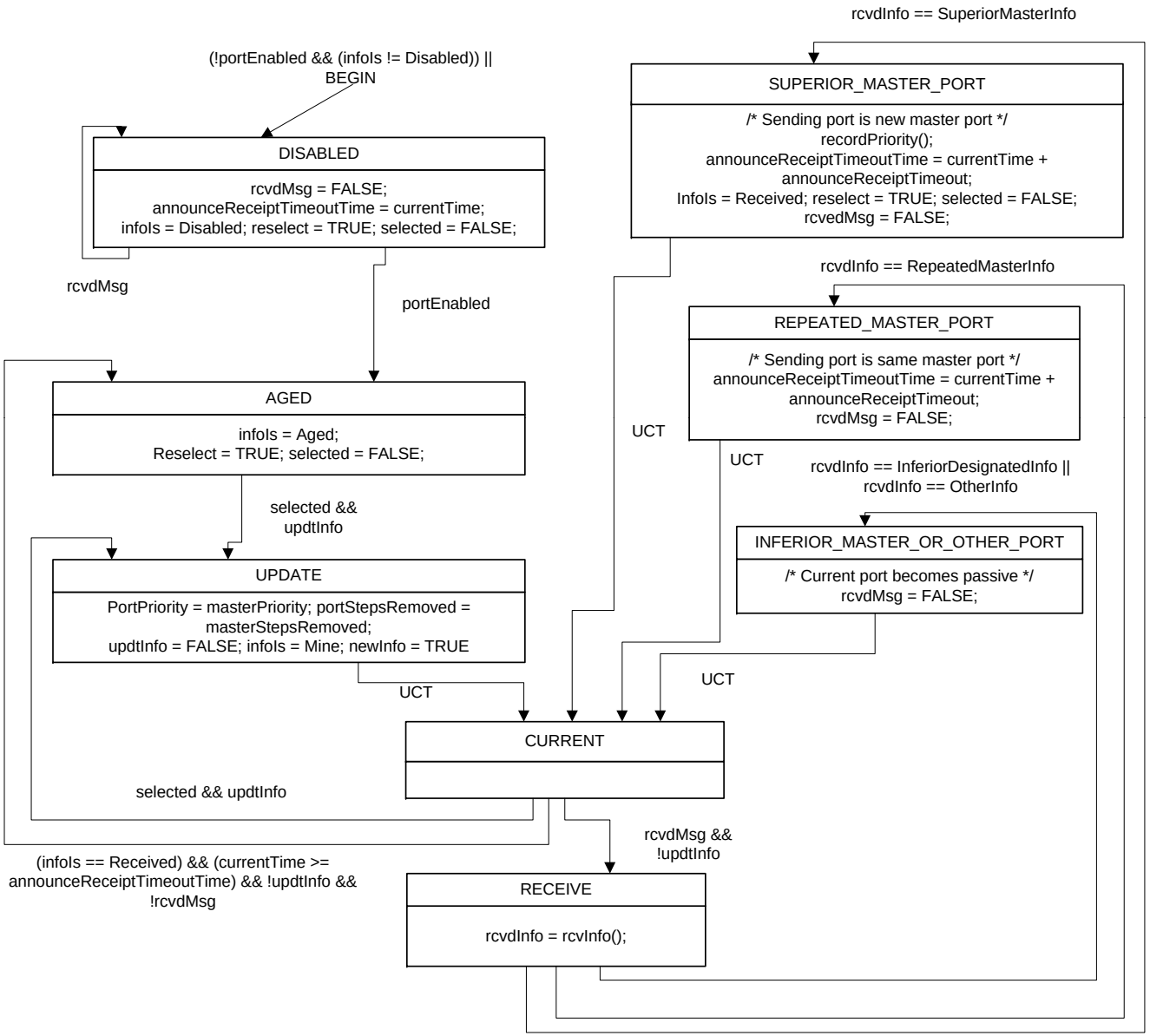
qualifyAnnounce() function - 3

- With the above as background, a port on an 802.1AS bridge or end station will do the following on receipt of an Announce message
  - If the message was sent by the same bridge, it is ignored
  - If the stepsRemoved field >= MAX_STEPS_REMOVED, it is ignored
    - MAX_STEPS_REMOVED = 255 in IEEE 1588
  - IEEE 1588 also checks that the Announce message is the most recently received Announce from the port that sent the message, by checking the sequence number
    - It is not necessary to make this explicit check in 802.1AS because the 802.3 and 802.11 PDUs are delivered in the order in which they were sent
  - Next, the foreign master data set is checked to see whether there is an entry for the clockIdentity of the bridge that sent this Announce message
  - If the answer is NO:
    - An entry for the bridge/end station that sent this Announce message is added to the foreign master data set
    - The variable timeWindow is set to currentTime+foreignMasterTimeWindow
  - Continued on next slide

qualifyAnnounce() function - 4

- On receipt of an Announce message (cont.):
  - If the answer is YES:
    - If currentTime < timeWindow, the Announce message is qualified
    - If currentTime >= timeWindow, the Announce message is not qualified
    - The variable timeWindow is set to currentTime+foreignMasterTimeWindow (regardless of which of the previous two sub-bullet items is true)
- If the received Announce message is qualified, qualifyAnnounce() returns TRUE
- If the received Announce message is not qualified, qualifyAnnounce() returns FALSE
- Entries can also be removed from the foreign master data set when it reaches a specified size; 1588 allows this but does not require it
  - 1588 does specify a default minimum size for the foreign master data set
- **Question for the AVB TG: Would the above logic for qualifyAnnounce() be best indicated in text form (as above), or in another form (e.g., flowchart)?**

Port Information State Machine



Note:

- Entering AGED sets reselect to TRUE.
- This causes ROLE_SELECTION block of the Port Role Selection state machine to be invoked.
- That causes selected to be set to TRUE.
- That allows the UPDATE block here to be entered if updtRolesTree() of the Port Role Selection state Machine has set updtInfo

infoIs global variable

- infoIs is a per port global variable that takes on the following values
  - Received: the port has received information from the master port for this communication path (LAN) and announce receipt timeout has not occurred (i.e., the information is not aged out)
  - Mine: the port has received information from the slave port of the current bridge (i.e., the current port is a master port)
  - Aged: announce receipt timeout has occurred on the port
  - Disabled: the port is disabled

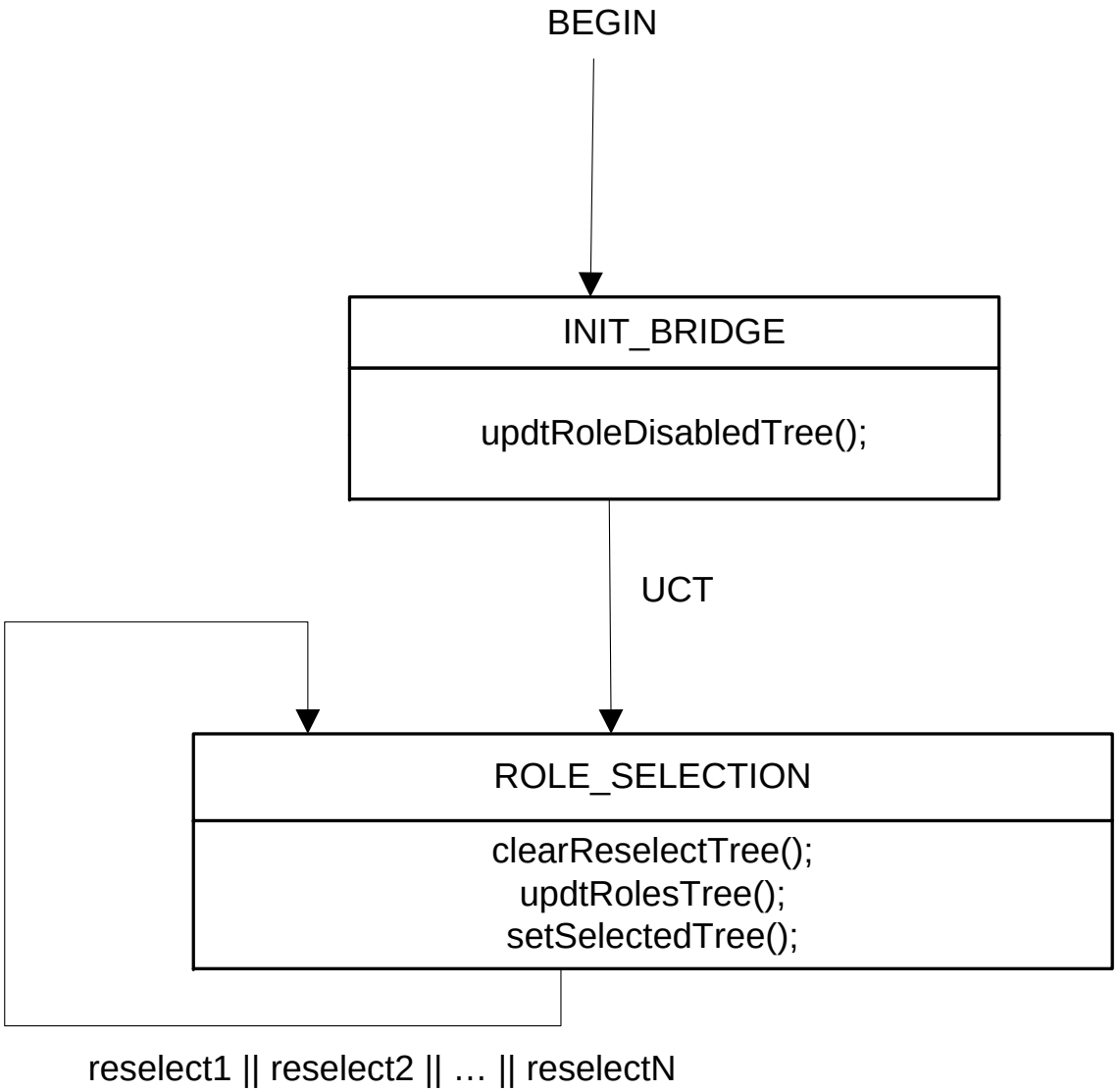
rcvInfo() function

- Corresponds to 1588 data set comparison algorithm
- Decodes the message priority vector and stepsRemoved field from the received Announce message, storing them in the msgPriority and msgStepsRemoved variables
- Returns SuperiorMasterInfo if the received message conveys a Master Port Role, and the message priority vector is superior to the Port's priority vector
  - In IEEE 1588, timer parameters (e.g., announce interval) do not influence whether a potential best master is better or worse than the current one
- Returns RepeatedMasterInfo if the received message conveys a Master Port Role and message priority vector is the same as the port's priority vector
- Returns InferiorMasterInfo if the received message conveys a Master Port Role and message priority vector that is worse than the port's priority vector
- Otherwise, returns OtherInfo

recordPriority() function

- Sets the components of the port priority vector equal to the components of the message priority vector

Port Role Selection State Machine



Several global variables - 1

- selectedRole is a per port global variable that indicates the port role
  - MasterPort
  - SlavePort
  - PassivePort
  - DisabledPort
- reselect is a per port global boolean variable
  - Setting reselect of any port to TRUE will cause the ROLE_SELECTION block of the Port Role Selection state machine to be re-entered, which in turn causes the roles of all the ports of the bridge to be updated (via the function updtRolesTree(), to be described shortly)
    - Setting reselect of any port to TRUE causes the equivalent of a 1588 state decision event
- selected is a per port global boolean variable
  - Selected is set to TRUE immediately after the roles of all the ports on the bridge are updated (via updtRolesTree()); this indicates to the Port Information state machine that it can update the port priority vector and other variables for each port

Several global variables - 2

- updtInfo is a per port global boolean variable set by the port role selection state machine
  - It indicates that the port information state machine should copy the newly determined master priority and masterStepsRemoved to port priority and portStepsRemoved, respectively

Several other functions

- updtRoleDisabledTree()
  - Sets the variable selectedRole to DisabledPort, for all ports on a bridge
- clearReselectTree()
  - Sets the variable reselect to FALSE, for all ports on a bridge
- setSelectedTree()
  - Sets the variable selected to TRUE, for all ports on a bridge, if the variable reselect is FALSE for all ports
  - If reselect is TRUE for any port, this function takes no action

updtRolesTree() function - 1

- Equivalent to 1588 computation of E_{best} using dataset comparison algorithm, followed by 1588 state decision algorithm
- The function first computes the following priority vectors
  a) GM path priority vector for each port that has a port priority vector and for which announce received timeout has not occurred
  b) GM priority vector for bridge, chosen as the best of the set of priority vectors consisting of the bridge priority vector (for this bridge) and all the calculated GM path priority vectors whose master port clockIdentity is not the clockIdentity of this bridge
  c) The first 4 components of the master priority vector for each port (i.e., all the components except the final portNumber)
- The function then computes masterStepsRemoved, which is equal to
  d) msgStepsRemoved for the port associated with the selected GM priority vector incremented by 1, if the GM priority vector is not the bridge priority vector
  e) 0 if the GM priority vector is the bridge priority vector

updtRolesTree() function - 2

- The function then assigns the port role for each port, and updates port priority vectors and timer information
  - f) If port is disabled (infoIs = Disabled), selectedRole set to DisabledPort; otherwise
  - g) If announce receipt timeout has occurred on the port (infoIs = Aged), updtInfo set to TRUE and selected Role set to MasterPort
  - h) If the port priority vector was derived from another port on the bridge or from the bridge itself as the GM (infoIs = Mine), selectedRole set to MasterPort. In addition, updtInfo is set to TRUE if the port priority vector differs from the master priority vector or portStepsRemoved differs from masterStepsRemoved
  - i) If the port priority vector was received in an Announce message and announce receipt timeout has not occurred (infoIs = Received), and the GM priority vector is now derived from it, selectedRole set to SlavePort and updtInfo is set to FALSE

updtRolesTree() function - 3

- Port Role assignment and port priority vector and timer information update (continued)

j) If the port priority vector was received in an Announce message and announce receipt timeout has not occurred (infoIs = Received), the GM priority vector is not now derived from it, the master priority vector is not higher than the port priority vector, and the master portIdentity component of the port priority vector (sourcePortIdentity) ***does not*** reflect another port on the bridge, selectedRole set to PassivePort and updtInfo is set to FALSE

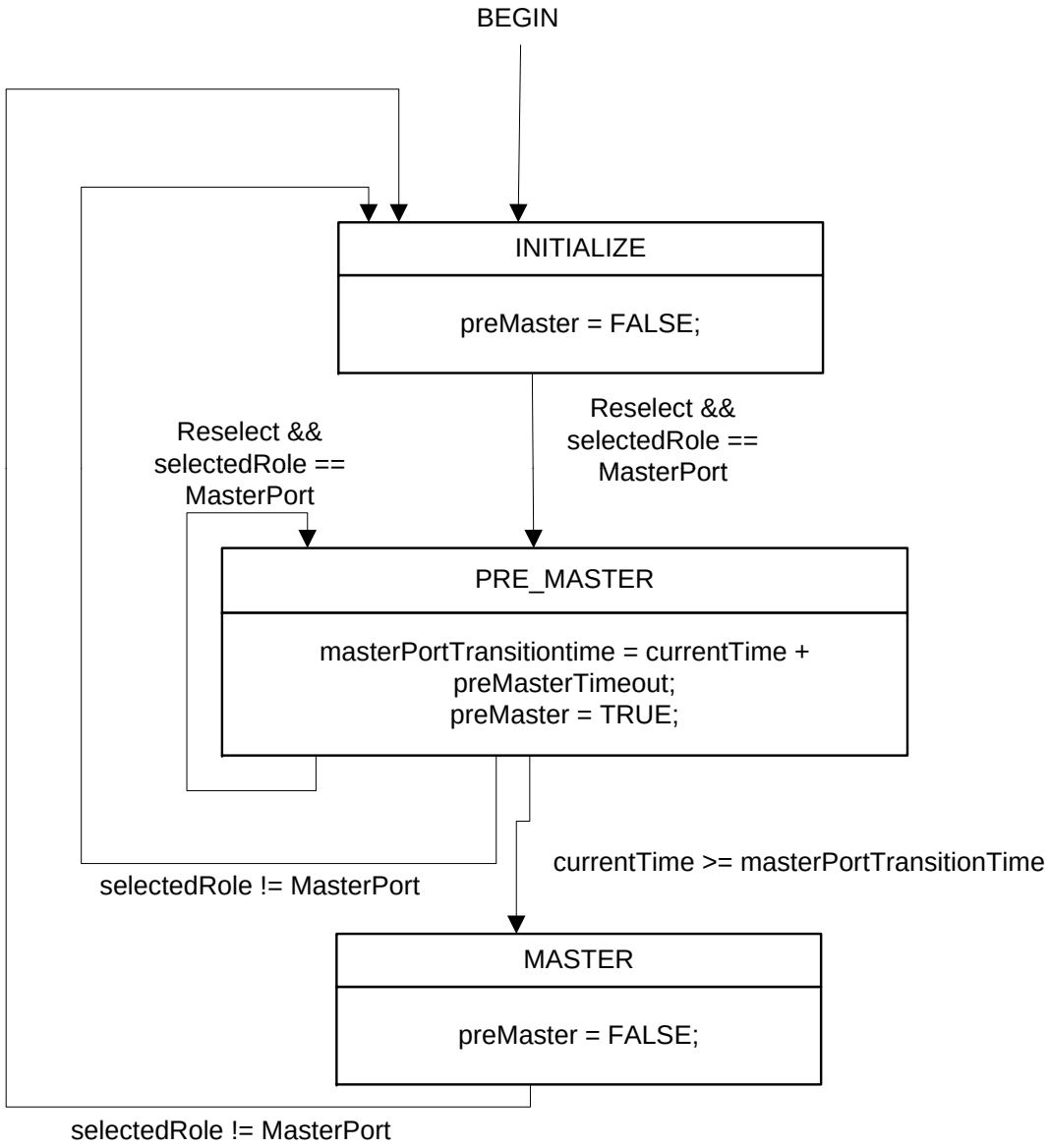
k) If the port priority vector was received in an Announce message and announce receipt timeout has not occurred (infoIs = Received), the GM priority vector is not now derived from it, the master priority vector is not higher than the port priority vector, and the master portIdentity component of the port priority vector (sourcePortIdentity) ***does*** reflect another port on the bridge, selectedRole set to PassivePort and updtInfo is set to FALSE

updtRolesTree() function - 4

- Port Role assignment and port priority vector and timer information update (continued)

l) If the port priority vector was received in an Announce message and announce receipt timeout has not occurred (infoIs = Received), the GM priority vector is not now derived from it, and the master priority vector is higher than the port priority vector, selectedRole set to MasterPort and updtInfo is set to TRUE

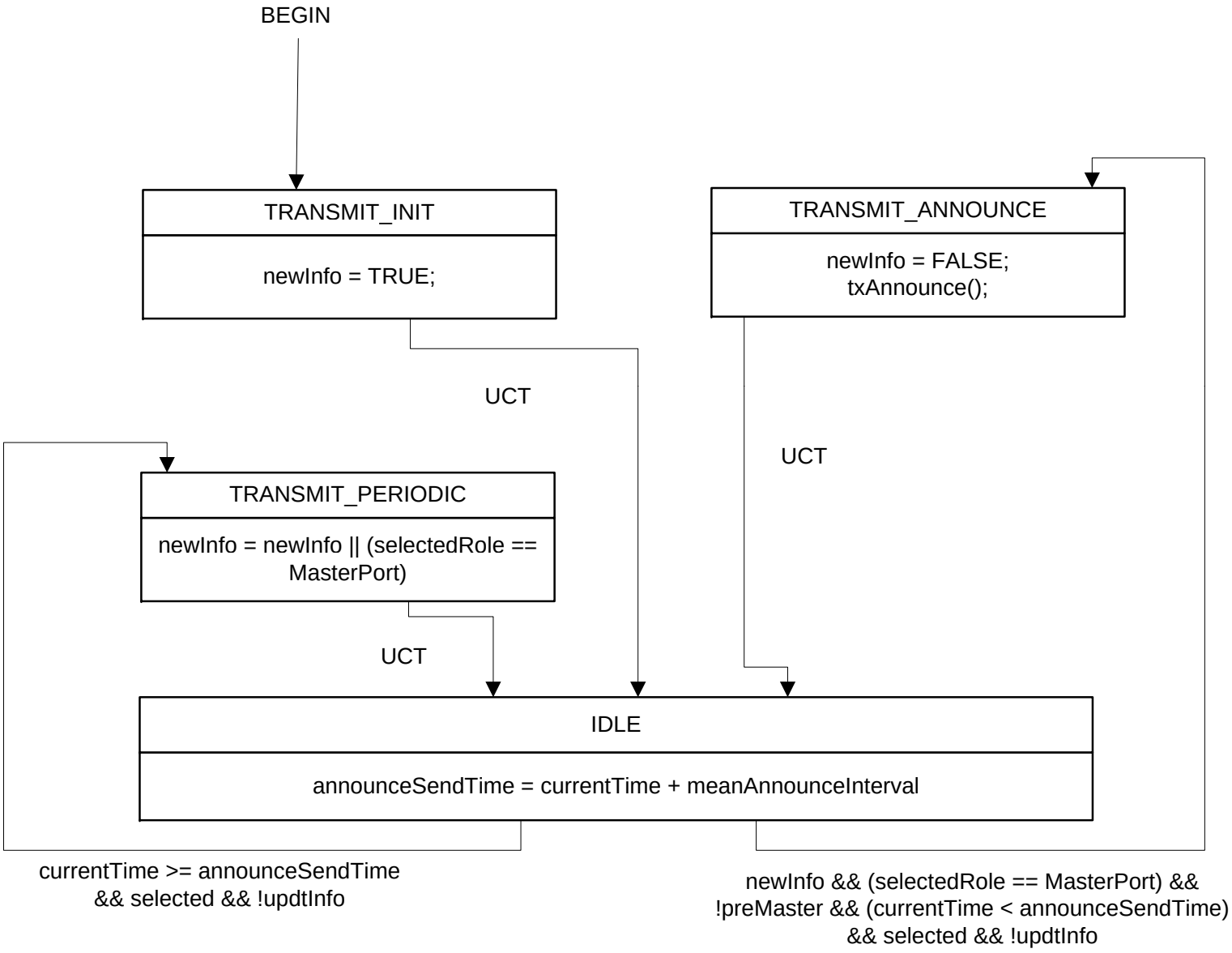
Pre-Master State Machine



preMasterTimeout

- preMasterTimeout is chosen so that preMaster remains true for the port (i.e., in 1588 the port remains in the PRE_MASTER state) long enough so any other better master that might be present has time to be heard from
- If masterStepsRemoved = 0
  - preMasterTimeout = 0
- Otherwise
  - preMasterTimeout = (masterStepsRemoved + 1)*meanAnnounceInterval

Port Transmit State Machine



txAnnounce() function

- Transmits an Announce message
  - The components of the message priority vector conveyed in the Announce message are set to the values of the respective components of the master priority vector for this port
  - The value of the stepsRemoved field of the Announce message is set equal to masterStepsRemoved for the port

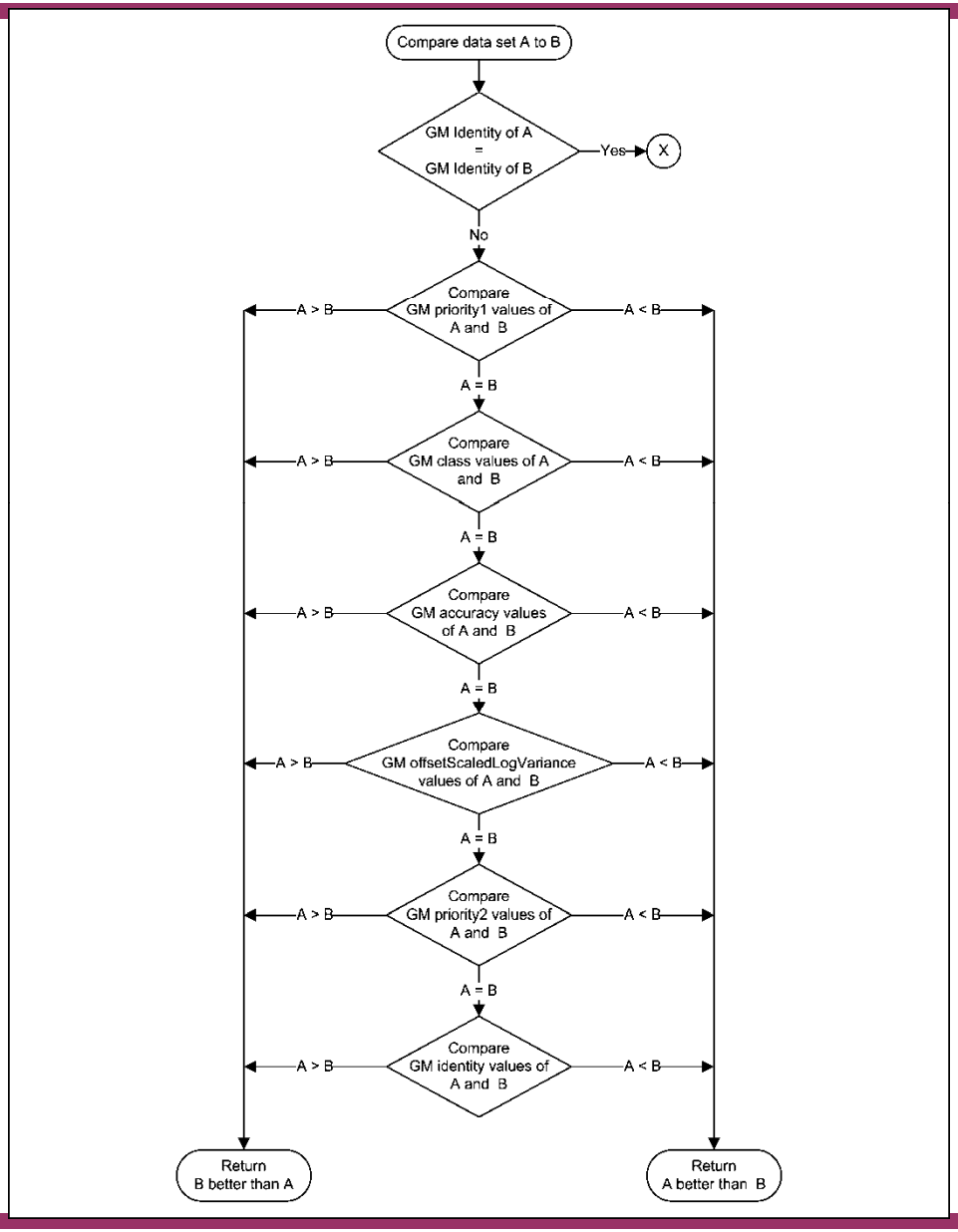
Examples - 1

- Two examples of best master selection follow, showing that the same result is obtained using the IEEE 1588 formulation and the RSTP-based formulation in each case
- Both examples are for cases where all bridges have the same clock attributes (priority1, clockClass, clockAccuracy, offsetScaledLogVariance, priority2), and the synchronization hierarchy (spanning tree) is chosen based on pathCost, clockIdentity, and portNumber
  - For simplicity, hop count is used for pathCost here
  - Note that the 1588 formulation uses hop count, but can be generalized for other pathCosts (e.g., sum of squares of link costs)
- In both examples, the GM is the node with the smallest clockIdentity, and the spanning tree is determined based on path cost

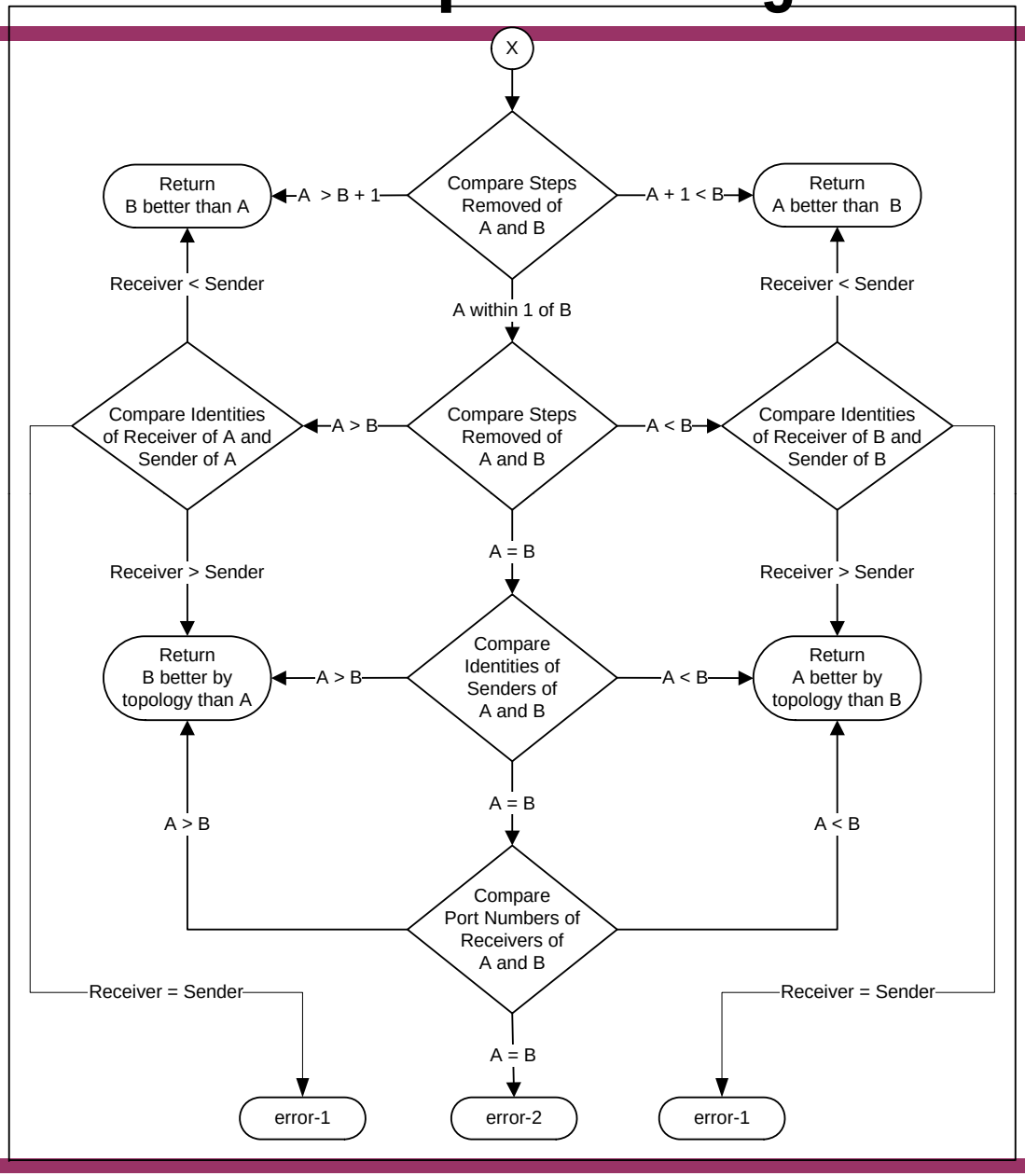
Examples - 2

- Example 1 illustrates the case of an odd number of nodes in a ring, with two nodes furthest from the GM
  - For the link connecting these nodes, one port is master and the other passive
- Example 2 illustrates the case of an even number of nodes in a ring, with one node furthest from the GM
  - For this node, one port is slave and the other port is passive
- The next 3 slides repeat the 1588 BMCA data set comparison and state decision algorithms, for convenience
  - These were shown earlier

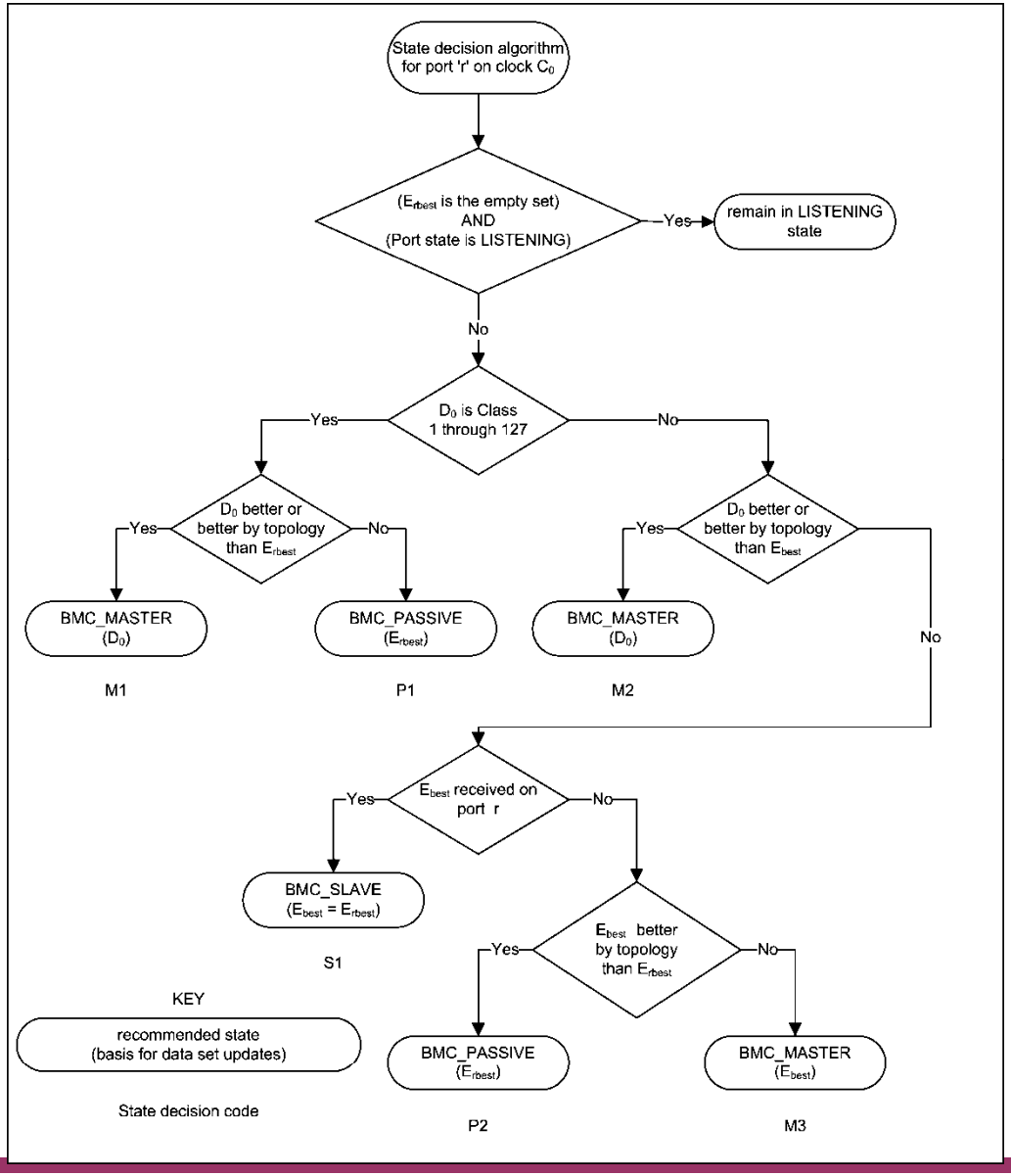
1588 Data Set Comparison Algorithm – 1



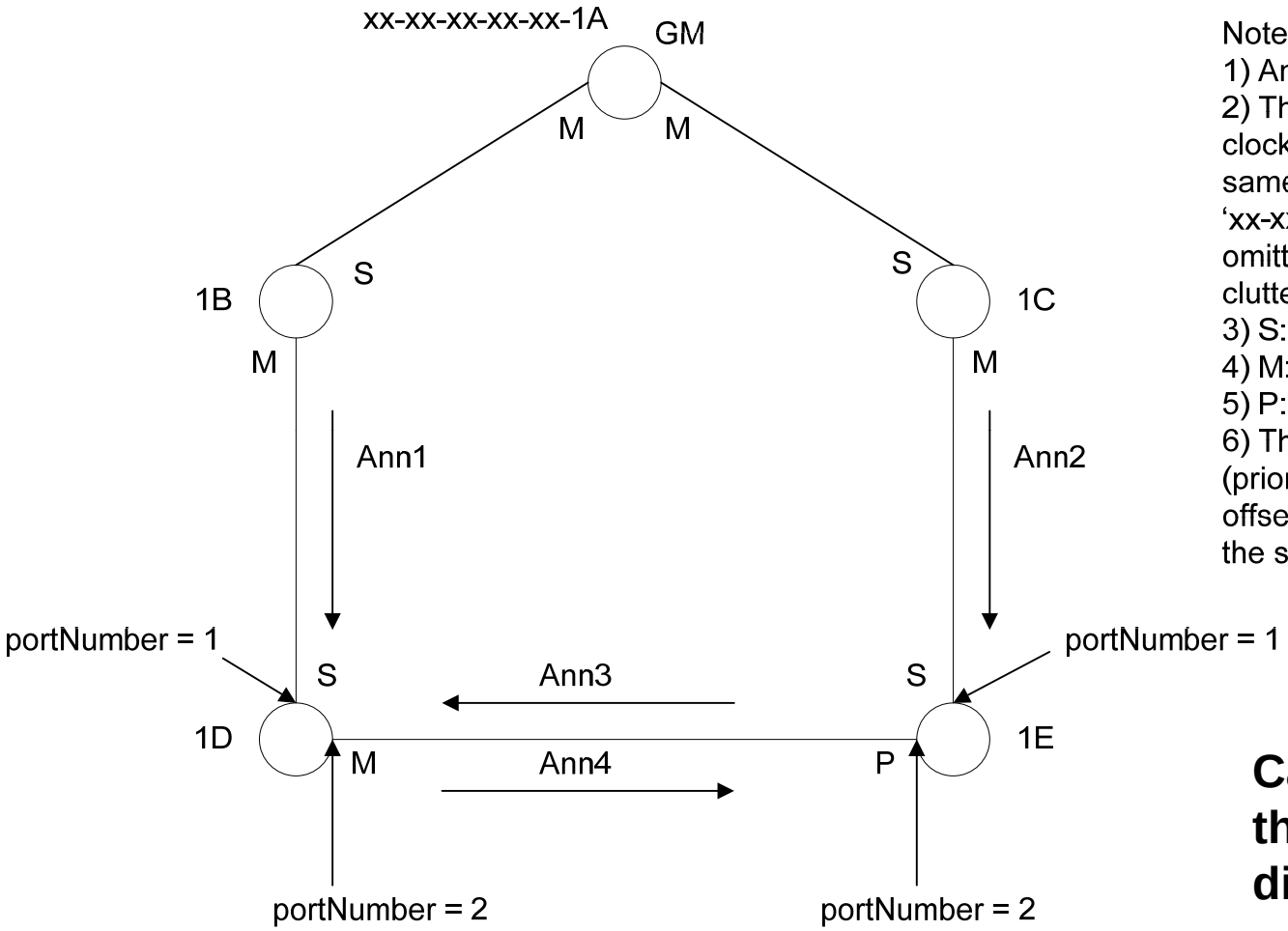
1588 Data Set Comparison Algorithm – 2



1588 State Decision Algorithm



Example 1 Configuration



Notes:
1) Annj: Announce message j
2) The most significant 40 bits of the clockIdentity of all 5 bridges are the same, and are indicated for the GM as 'xx-xx-xx-xx-xx'. This indication is omitted for the other 4 bridges to reduce clutter in the figure.
3) S: slave port
4) M: master port
5) P: passive port
6) The attributes other than clockIdentity (priority1, clockClass, clockAccuracy, offsetScaledLogVariance, priority2) are the same for all the bridges

Case of 2 paths to the GM with different pathCosts

Example 1, Bridge 1D, 1588 Formulation

- stepsRemoved (Ann1) = 2 (on port 1)
- stepsRemoved (Ann3) = 3 (on port 2)
- Therefore, stepsRemoved for Ann1 and Ann3 is within 1 when comparing them (2nd slide of data set comparison algorithm), with stepsRemoved for Ann1 smaller
- Compare clockIdentity of receiver of Ann3 and sender of Ann3
- clockIdentity (sender of Ann3) = 1E > clockIdentity (receiver of Ann3) = 1D
- Therefore, Ann1 is 'better' than Ann3 (as opposed to 'better by topology')
- Therefore, port 2 of bridge 1D becomes a master port (point M3 of state decision algorithm)

Example 1, Bridge 1D, RSTP-based Formulation

- GM priority vector = {R_{1A}: 2: {1B: 2}: 1}
- Master priority vector at port 2 = {R_{1A}: 2: {1D: 2}: 2}
- Port priority vector at port 2 of 1D is based on Announce message received from 1E (message priority vector); port priority vector = {R_{1A}: 2: {1E: 2}: 2}
  - Note that the path cost of the final hop is not added for the message priority vector
- Master priority vector at port 2 < port priority vector at port 2
- Therefore, port 2 of 1D is a master port, based on decision (l) of updtRolesTree()

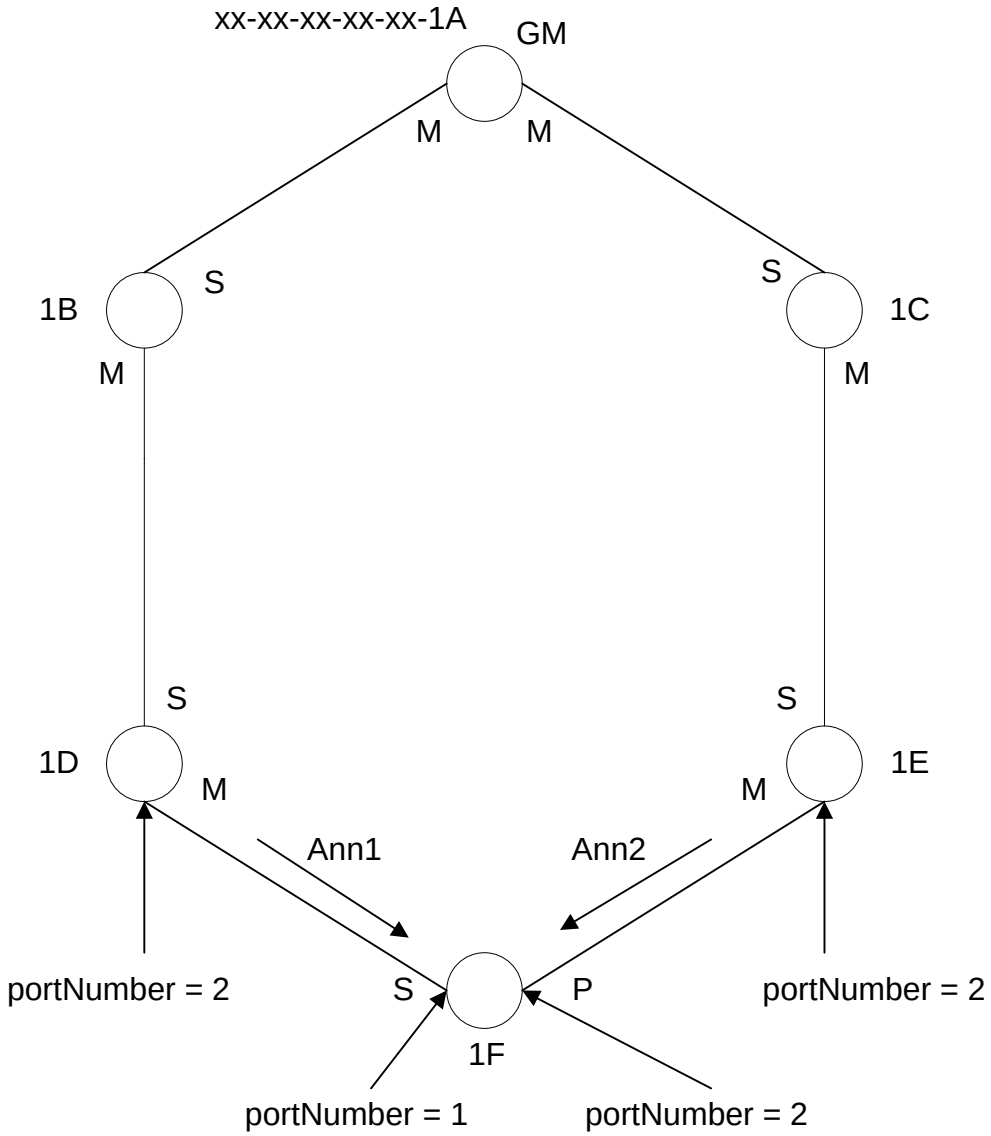
Example 1, Bridge 1E, 1588 Formulation

- stepsRemoved (Ann2) = 2 (on port 1)
- stepsRemoved (Ann4) = 3 (on port 2)
- Therefore, stepsRemoved for Ann2 and Ann4 is within 1 when comparing them (part 2 of data set comparison algorithm), with stepsRemoved for Ann2 smaller
- Compare clockIdentity of receiver of Ann4 and sender of Ann4
- clockIdentity (sender of Ann4) = 1D < clockIdentity (receiver of Ann4) = 1E
- Therefore, Ann1 is ‘better by topology’ than Ann3 (as opposed to ‘better’)
- Therefore, port 2 of bridge 1D becomes a passive port (point P2 of state decision algorithm)

Example 1, Bridge 1E, RSTP-based Formulation

- GM priority vector = {R_{1A}: 2: {1C: 2}: 1}
- Master priority vector at port 2 = {R_{1A}: 2: {1E: 2}: 2}
- Port priority vector at port 2 of 1E is based on Announce message received from 1D (message priority vector); port priority vector = {R_{1A}: 2: {1D: 2}: 2}
  - Note that the path cost of the final hop is not added for the message priority vector
- Master priority vector at port 2 > port priority vector at port 2
- Therefore, port 2 of 1D is a passive port, based on decision (j) of updtRolesTree()

Example 2 Configuration



Case of 2 paths to the GM with the same pathCost

Notes:
1) Annj: Announce message j
2) The most significant 40 bits of the clockIdentity of all 6 bridges are the same, and are indicated for the GM as 'xx-xx-xx-xx-xx'. This indication is omitted for the other 5 bridges to reduce clutter in the figure.
3) S: slave port
4) M: master port
5) P: passive port
6) The attributes other than clockIdentity (priority1, clockClass, clockAccuracy, offsetScaledLogVariance, priority2) are the same for all the bridges

Example 2, Bridge 1F, 1588 Formulation - 1

- 1588 computes E_{best} at 1F = best of (Ann1, Ann2, 1F local clock)
  - This is done for the bridge as a whole (i.e., this is not a per port computation)
- Ann1 and Ann2 are both better than 1F, because they reflect bridge 1A (the GM)
- Ann1 and Ann2 both have stepsRemoved = 3
- Compare clockIdentity of sender of Ann1 with clockIdentity of sender of Ann2 (data set comparison algorithm, part 2)
- Sender of Ann1 = 1D
- Sender of Ann2 = 1E
- Sender of Ann1 < sender of Ann2
- Therefore, Ann1 is 'better by topology' than Ann2
- Also, E_{best} = Ann1
- Next, invoke state decision algorithm at each port of 1F

Example 2, Bridge 1F, 1588 Formulation - 2

- State decision algorithm at 1F, port 1
- E_{best} = Ann1, and was received on port 1
- Therefore, port 1 of 1F becomes a slave port (point S1 of state decision algorithm)

- State decision algorithm at 1F, port 2
- E_{best} = Ann1, and was not received on port 2
- E_{best} is better by topology than the best announce received on port 2 (which is Ann2)
- Therefore, port 2 of 1F becomes a passive port (point P2 of state decision algorithm)

Example 2, Bridge 1F, RSTP-based Formulation

- GM path priority vector on port 1 = {R_{1A}: 3: {1D: 2}: 1}
- GM path priority vector on port 2 = {R_{1A}: 3: {1E: 2}: 2}
- Bridge priority vector for 1F = {R_{1F}: 0: {1F: 0}: 0}
- GM priority vector for bridge 1F = best of the above 3 priority vectors = {R_{1A}: 3: {1D: 2}: 1} (received on port 1)
- Therefore, the GM priority vector is derived from port 1
- Therefore, port 1 is a slave port, based on decision (i) of updtRolesTree()
- Next, look at port 2 of 1F
- Master priority vector at port 2 = {R_{1A}: 3: {1F: 2}: 2}
- Port priority vector at port 2 = {R_{1A}: 2: {1E: 2}: 2}, based on message priority vector represented by Ann2
- Therefore, port priority vector at port 2 < master priority vector at port 2
- Therefore, port 2 of 1F is a passive port, based on decision (j) of updtRolesTree()

802.1AS Entities - 1

- In the formalism in this presentation, which follows the formalism used for RSTP in 802.1D-2004, the state machines operate in the 802.1AS SiteSync and PortSync entities
- Communication among the state machines is done via global variables
- In this approach, it appears there is no reason to specifically refer to the SiteSync and PortSync entities explicitly
  - Rather, there are simply per bridge and per port state machines (and the instances of corresponding functions and variables)
  - Note that the SiteSync and PortSync entities were not referred to in this presentation (there was no need)
  - It will likely be convenient to refer to a single media-independent entity that encompasses all the per bridge and per port state machines, functions, and variables

802.1AS Entities - 2

- It will be necessary, however, to refer to the TS entities (one per port), because these are different for different media
  - The TS entity for each medium takes the Announce information from the media-independent entity (in a request primitive) and transmits the information in a PDU to a peer entity at the other end of the link
  - Likewise, the TS entity for each medium receives a PDU (from a peer entity at the other end of the link) that contains Announce information and presents the information to the media-independent entity (in an indication primitive)
- For synchronization information, the service interface primitives will be different for different media (as in 802.1AS D2.0)

References

1. Norman Finn, *802.1AS Fast Master Clock Selection, Moving 802.1AS closer to RSTP*, Version 2, April, 2008.